Python Programming

Regression Analysis

Dr. Chun-Hsiang Chan
Department of Geography
National Taiwan Normal University

Outlines

- Scaling Problems
- Ordinary Least Squares
- Lasso Regression
- Ridge Regression
- Generalized Linear Model
- Evaluation Metrics
- Assignments

Scaling Problems

- Typically, we have five numerical transformation methods.
  1. *Normalization*
  2. *Standardization*
  3. *Binarization*
  4. *Centralization*
  5. *Feature scaling*
- Among these five methods, standardization is the most common method in statistical analysis because of its range of transformed value.

Scaling Problems

Let $p \geqq 1$ be a real number.

The $p - norm$ (= $\ell_p - norm$) of vector $x = (x_1, \ldots, x_n)$ is

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Given $p = 1$

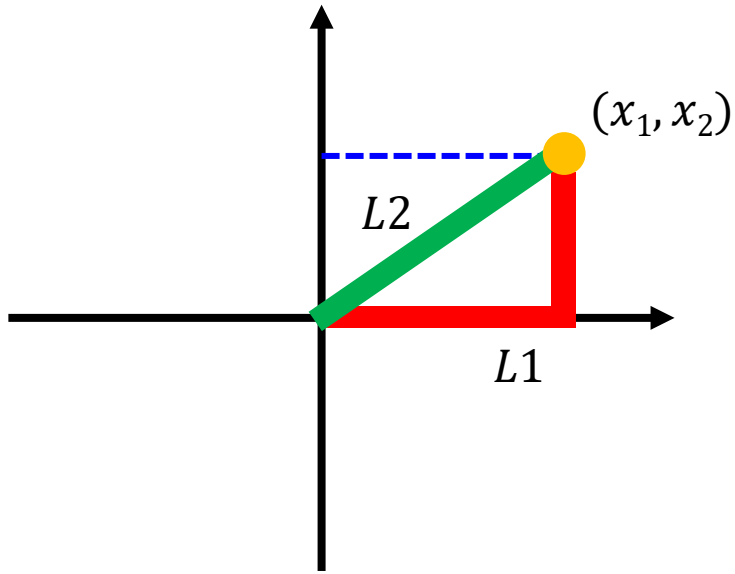
$$L1 = \|x\|_1 = |x_1| + |x_2|$$

Given $p = 2$

$$L2 = \|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2}$$

Given $p = \infty$

$$L\infty = \|x\|_\infty = \max(x)$$



Scaling Problems

Methods	Equations
Standardization	$x' = \frac{x - \mu}{\sigma}$
Binarization	$if\ x \geq threshold; x' = 1$ $else\ x' = 0$
Centralization	$x' = x - \frac{1}{n}\sum_{i=1}^{n} x_i$
Feature scaling	$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$

μ and σ denote mean and standard deviation, respectively.

Regression Analysis

- **Objective function:**
  - **Linear Regression:**

$$Min_w \frac{1}{m} \sum_i (y_i - w^T x_i)^2$$

  - **Lasso Regression:**

$$Min_w \frac{1}{m} \left(\sum_i (y_i - w^T x_i)^2 + \lambda \sum_j^n |w_j| \right)$$

  - **Ridge Regression:**

$$Min_w \frac{1}{m} \left(\sum_i (y_i - w^T x_i)^2 + \lambda \sum_j^n w_j^2 \right)$$

Ordinary Least Squares (OLS)

- **Ordinary Least Squares (OLS)** is a statistical method used to estimate the parameters of a linear regression model by minimizing the sum of squared differences between observed and predicted values.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i, i = 1, \ldots, n$$

$$y = X\beta + \varepsilon$$

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

- **Simplicity:** Provides a straightforward and interpretable model.
- **Optimality:** Under the Gauss-Markov assumptions, OLS estimates are BLUE (Best Linear Unbiased Estimators).
- **Foundation for Inference:** Allows construction of confidence intervals and hypothesis tests.

Assumptions of OLS

- The hypothesis of residuals in the OLS:
  (1) The expected value of residuals is zero (zero mean)
  $E(u) = 0$
  (2) Residuals follow homoskedasticity
  $var(u) = \sigma^2, \sigma^2 \; is \; a \; constant$
  (3) Residuals are non-autocorrelation
  $cov(ut, ut - s) = 0, for \; s \neq 0$
  (4) The relationship between independent variables and residuals is orthogonality
  $cov(x, u) = 0, for \; any \; I$
  (5) Residuals follow normality

  The residuals follow abovementioned conditions can be regards as independently identical distribution (iid) residuals: $u \sim i.i.d.N(0, \sigma^2)$

OLS – Mathematical Derivation

- We have n observations and for each sample consists of

$$observation = (x_i, y_i), i = 1,2, \dots, n$$

where $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ is a vector of p predictors (features/ independent variables) and y_i is the response (dependent variables).

- OLS fits a linear model

$$\hat{y}_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

- Using a matrix representation way, …

$$y = X\beta + \varepsilon$$

where $y \in \mathbb{R}^n$ is the vector of all dependent response

$X \in \mathbb{R}^{n \times (p+1)}$ is the predictor matrix (first column is the intercept)

$\beta \in \mathbb{R}^{(p+1)}$ is the vector of parameters: $\beta_0, \beta_1, \beta_2, \dots, \beta_p$

ε is the residual vector.

OLS – Mathematical Derivation

- **Objective functions:** sum of squared error (SSE)
  OLS estimates β by minimizing the sum of squared residuals (errors):

$$SSE(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i) = \sum i = 1^n \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- In matrix notation, …

$$SSE(\beta) = (y - X\beta)^T (y - X\beta)$$

OLS – Assumptions

- **Assumptions for Best Linear Unbiased Estimator (BLUE)**
- Under the **Gauss-Markov assumptions** (linear model, no perfect collinearity, zero mean error, errors have constant variance σ^2, and errors uncorrelated), the OLS estimator:
- $\hat{\beta}$ is the **Best Linear Unbiased Estimator** (BLUE)—i.e., it has the smallest variance among all unbiased linear estimators.

OLS – Solving Coefficient Steps

- **Solution via Normal Equations**
  1) To find the minimum, we take the derivative of SSE with respect to β and set it to zero

$$\frac{\partial}{\partial \beta}[(y - X\beta)^T(y - X\beta)] = 2X^T(y - X\beta)$$

  2) Normal Equations

$$X^T X\beta = X^T y$$

  3) Closed-Form Solution

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

  4) Provided $X^T X$ is invertible (i.e., X has full column rank)

OLS – Coefficient (1)

- **Solution via Normal Equations**
  1) To find the minimum, we take the derivative of SSE with respect to β and set it to zero

$$\frac{\partial}{\partial \beta} S(\beta) = \frac{\partial}{\partial \beta} [(y - X\beta)^T (y - X\beta)]$$

$$= \frac{\partial}{\partial \beta} [(y^T - \beta^T X^T)(y - X\beta)]$$

$$= \frac{\partial}{\partial \beta} [y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta]$$

$\because y^T X\beta$ is a scaler, $\therefore (y^T X\beta)^T = \beta^T X^T y$

$$= \frac{\partial}{\partial \beta} [y^T y - 2\beta^T X^T y + \beta^T X^T X\beta]$$

OLS – Coefficient (1)

$$\frac{\partial}{\partial \beta}[y^T y - 2\beta^T X^T y + \beta^T (X^T X)\beta]$$

$$\text{take } \nabla_\beta S(\beta) \begin{cases} \frac{\partial}{\partial \beta}(y^T y) = 0 & constant\ in\ \beta \\ \frac{\partial}{\partial \beta}(-2\beta^T X^T y) = -2X^T y & linear\ form \\ \frac{\partial}{\partial \beta}(\beta^T (X^T X)\beta) = 2(X^T X)\beta & quadratic\ term, X^T X \in sysmtric \end{cases}$$

$$\nabla_\beta S(\beta) = 0 - 2X^T y + 2(X^T X)\beta = 2X^T (y - X\beta)$$

OLS – Coefficient (2) & (3)

2) Normal Equations

$$\min_{\beta} 2X^T(y - X\beta)$$

$$\Rightarrow 2X^T(y - X\beta) = 0$$

$$X^T y = X^T X\beta$$

3) Closed-Form Solution

$$\therefore \hat{\beta} = (X^T X)^{-1} X^T y$$

OLS – 95% C.I. & T-test :: Residuals

- **Residuals & Residual Variance**

  1) Residuals:

$$\hat{e} = y - X\hat{\beta}$$

  2) Residual Sum of Squares (RSS or SSE):

$$RSS = \hat{e}^T\hat{e} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  3) Estimate of Error Variance $\hat{\sigma}^2$ (a.k.a. MSE)

$$\hat{\sigma}^2 = \frac{RSS}{n-(p+1)}, degree\ of\ freedom = n-(p+1)$$

n is number of samples; p is number of predictors; 1 is constant.

OLS – 95% C.I. & T-test :: COV(Residuals)

- **Variance-Covariance Matrix of $\hat{\beta}$**
- From OLS theory, …

$$var(\hat{\beta}) \approx \hat{\sigma}^2 (X^T X)^{-1}$$

$$\hat{\beta} = (X^T X)^{-1} X^T y = (X^T X)^{-1} X^T (X\beta + \varepsilon) = \beta + (X^T X)^{-1} X^T \varepsilon$$

Since ε has variance $\sigma^2 I$

$$var(\hat{\beta}) = var[\beta + (X^T X)^{-1} X^T \varepsilon] = var[(X^T X)^{-1} X^T \varepsilon]$$

Since $(X^T X)^{-1} X^T$ is a constant matrix

$$var(\hat{\beta}) = (X^T X)^{-1} X^T var[\varepsilon] X \left((X^T X)^{-1}\right)^T$$

OLS – 95% C.I. & T-test :: COV(Residuals)

With $var(\varepsilon) = \sigma^2 I$ and noting $(X^T X)^{-1}$ is symmetric

$$var(\hat{\beta}) = (X^T X)^{-1} X^T (\sigma^2 I) X \left((X^T X)^{-1}\right)^T = \sigma^2 (X^T X)^{-1} X^T X \left((X^T X)^{-1}\right)^T$$
$$= \sigma^2 (X^T X)^{-1}$$

- Hence, in ideal theoretical form:

$$var(\varepsilon) = \sigma^2 (X^T X)^{-1}$$

- Replacing σ^2 with $\hat{\sigma}^2$

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n - (p+1)} \Rightarrow var(\hat{\beta}) = \hat{\sigma}^2 (X^T X)^{-1}$$

- Hence, the standard error of each coefficient $\hat{\beta}_j$ is

$$SE(\hat{\beta}_j) = \sqrt{\hat{\sigma}^2 [(X^T X)^{-1}]_{jj}}$$

OLS – 95% C.I. & T-test :: 95 % C.I. Est.

- A two-sided 95% confidence interval (C.I.) for $\hat{\beta}_j$ is given by

$$\hat{\beta}_j \pm t_{0.5\times\alpha,v} \times SE\left(\hat{\beta}_j\right)$$

where $t_{0.5\times\alpha,v}$ is the critical value from the Student's t-distribution with $v = n - (p + 1)$ degrees of freedom, for $\alpha = 0.05$ (i.e., 95% C.I.)

$\alpha = 0.05$ means $t_{0.025,v}$ because it's a two-tailed test with 2.5% in each tail

OLS – 95% C.I. & T-test :: T-test for β

- The null hypothesis of each predictor follows:

$$\begin{cases} H_0: \beta_j = 0 \\ H_1: \beta_j \neq 0 \end{cases}$$

- The t-statistics is:

$$t_j = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

- The correponding *p-value* is:

$$p - value = 2 \times \left[1 - F_v(|t_j|)\right]$$

where F_v is the CDF of the Student's t-distribution with $v = n - (p + 1)$

Regression Analysis :: OLS

```
from sklearn.datasets import load_diabetes
import statsmodels.api as sm
from scipy import stats
print(load_diabetes(as_frame=True).DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

:Number of Instances: 442

:Number of Attributes: First 10 columns are numeric predictive values

:Target: Column 11 is a quantitative measure of disease progression one year after baseline

:Attribute Information:
    - age     age in years
    - sex
    - bmi     body mass index
    - bp      average blood pressure
    - s1      tc, total serum cholesterol
    - s2      ldl, low-density lipoproteins
    - s3      hdl, high-density lipoproteins
    - s4      tch, total cholesterol / HDL
    - s5      ltg, possibly log of serum triglycerides level
    - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard
deviation times the square root of `n_samples` (i.e. the sum of squares of each column tota
ls 1).

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html
```

Regression Analysis :: OLS

```
# fetch X, y data
dfx = load_diabetes(as_frame=True).data
dfy = load_diabetes(as_frame=True).target
# preview dataset
dfx.head()
```

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641

Regression Analysis :: OLS

```
# standardized x
x_scaled = stats.zscore(dfx.values, axis=0, ddof=1)
# add constant
x_scaled = sm.add_constant(x_scaled)
# OLS fitting
OLS_reg = sm.OLS(dfy, x_scaled).fit()
# preview summary
OLS_reg.summary(xname=['const']+list(dfx.columns))
```

Regression Analysis :: OLS

```
# preview summary
OLS_reg.summary(xname=['const']+list(dfx.columns))
```

OLS Regression Results

Dep. Variable:	target	**R-squared:**	0.518
Model:	OLS	**Adj. R-squared:**	0.507
Method:	Least Squares	**F-statistic:**	46.27
Date:	Tue, 04 Feb 2025	**Prob (F-statistic):**	3.83e-62
Time:	03:09:15	**Log-Likelihood:**	-2386.0
No. Observations:	442	**AIC:**	4794.
Df Residuals:	431	**BIC:**	4839.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	152.1335	2.576	59.061	0.000	147.071	157.196
age	-0.4767	2.845	-0.168	0.867	-6.069	5.116
sex	-11.4198	2.915	-3.917	0.000	-17.150	-5.690
bmi	24.7546	3.168	7.813	0.000	18.527	30.982
bp	15.4469	3.115	4.958	0.000	9.324	21.570
s1	-37.7226	19.842	-1.901	0.058	-76.722	1.276
s2	22.7019	16.144	1.406	0.160	-9.030	54.433
s3	4.8116	10.121	0.475	0.635	-15.080	24.703
s4	8.4316	7.689	1.097	0.273	-6.682	23.545
s5	35.7749	8.186	4.370	0.000	19.686	51.864
s6	3.2203	3.142	1.025	0.306	-2.955	9.396

Omnibus:	1.506	**Durbin-Watson:**	2.029
Prob(Omnibus):	0.471	**Jarque-Bera (JB):**	1.404
Skew:	0.017	**Prob(JB):**	0.496
Kurtosis:	2.726	**Cond. No.**	21.7

Regression Analysis :: Lasso

- **Lasso (Least Absolute Shrinkage and Selection Operator)** is a linear regression method that includes an **L1 penalty** on the coefficients, encouraging some to become exactly zero. This makes Lasso both a **regularization** technique (to avoid overfitting) and a **feature selection**.

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \sum_{j=1}^{p} |\beta_j|$$

- The **L1 penalty** term $\lambda \sum_{j=1}^{p} |\beta_j|$ exerts **absolute shrinkage**, driving many β_j to zero.
- λ (often called α in some softwares, e.g., scikit-learn) controls the penalty strength.

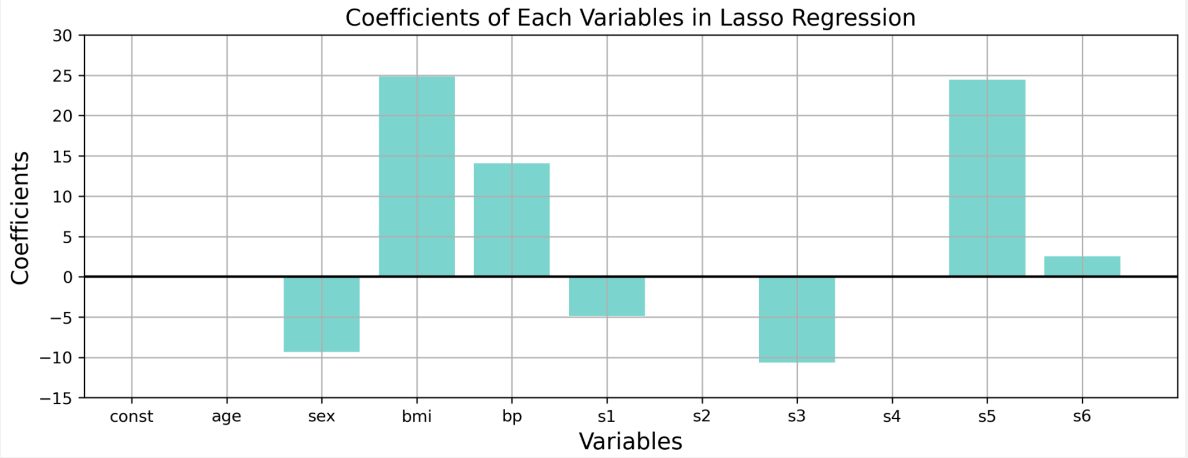
Regression Analysis :: Lasso

- **Features:**
  1) Sparse solution: Forces many coefficients to be zero.
  2) Feature selection: Automatically picks out the most important predictors. Especially in high-dimensional data, you suspect many coefficients should be zero.
- **Pros:**
  1) Produces interpretable models by eliminating unnecessary features
  2) Helps when p (number of predictors) is very large
- **Cons:**
  1) Biased estimates due to absolute penalty
  2) If features are highly correlated, Lasso may select one arbitarily and ignore the others

Regression Analysis :: Lasso

```
from sklearn.linear_model import Lasso
lassoReg = Lasso(alpha=1)
lassoReg.fit(x_scaled, dfy)

plt.figure(figsize=[12,4], dpi=300)
plt.bar(np.arange(x_scaled.shape[1]), lassoReg.coef_, facecolor='#81D8D0')
plt.xticks(np.arange(x_scaled.shape[1]), ['const']+list(dfx.columns), rotation=0)
plt.plot([-0.5, x_scaled.shape[1]], [0, 0], 'k')
plt.title('Coefficients of Each Variables in Lasso Regression', fontsize=14)
plt.xlabel('Variables', fontsize=14)
plt.ylabel('Coefficients', fontsize=14)
plt.axis([-0.5, x_scaled.shape[1], -15, 30])
plt.grid(True)
plt.show()
```



Regression Analysis :: Ridge

- **Ridge Regression** is a linear regression method that includes an **L2 penalty** term to shrink the magnitude of coefficients. This addresses multicollinearity and helps reduce overfitting.

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- The **L2 penalty** term $\lambda \sum_{j=1}^{p} \beta_j^2$ applies a squared shrinkage on coefficients.
- λ (sometimes called α in software like scikit-learn) controls the strength of regularization.

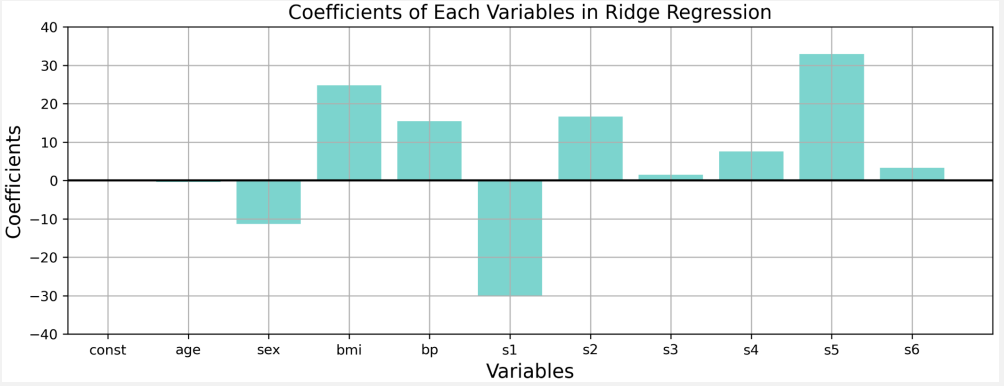
Regression Analysis :: Ridge

- **Pros:**
  - Collinearity handling: mitigates issues when predictors are highly correlated.
  - Overfitting control: reduces model variance at the cost of a small bias increase.
  - Prefers a smoother/ more stable predictions, not strict sparsity
  - Works well with many correlated features
  - Stabilizes coefficient estimates
  - Often improves prediction accuracy over plain OLS when p is large
- **Cons:**
  - Does not perform feature selection (no coefficients are driven fully to zero)
  - Choosing λ requires cross-validation (adds computation)

Regression Analysis :: Ridge

```
from sklearn.linear_model import Ridge
ridgeReg = Ridge(alpha=1)
ridgeReg.fit(x_scaled, dfy)

plt.figure(figsize=[12,4], dpi=300)
plt.bar(np.arange(x_scaled.shape[1]), ridgeReg.coef_, facecolor='#81D8D0')
plt.xticks(np.arange(x_scaled.shape[1]), ['const']+list(dfx.columns), rotation=0)
plt.plot([-0.5, x_scaled.shape[1]], [0, 0], 'k')
plt.title('Coefficients of Each Variables in Ridge Regression', fontsize=14)
plt.xlabel('Variables', fontsize=14)
plt.ylabel('Coefficients', fontsize=14)
plt.axis([-0.5, x_scaled.shape[1], -40, 40])
plt.grid(True)
plt.show()
```



Regression Analysis :: GLM

When your data does not follow normal distribution, you must find a suitable distribution for regression analysis.

Moreover, if your data has many zeros, you must use the zero-inflated version of GLM to overcome the zero inflation problem.

Common distributions with typical uses and canonical link functions

Distribution	Support of distribution	Typical uses	Link name	Link function, $\mathbf{X}\boldsymbol{\beta} = g(\mu)$	Mean function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$	$\mu = \mathbf{X}\boldsymbol{\beta}$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Negative inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$	$\mu = -(\mathbf{X}\boldsymbol{\beta})^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = \mu^{-2}$	$\mu = (\mathbf{X}\boldsymbol{\beta})^{-1/2}$
Poisson	integer: $0, 1, 2, \ldots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$	$\mu = \exp(\mathbf{X}\boldsymbol{\beta})$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$
Binomial	integer: $0, 1, \ldots, N$	count of # of "yes" occurrences out of N yes/no occurrences		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{n-\mu}\right)$	
Categorical	integer: $[0, K)$ *K*-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1	outcome of single *K*-way occurrence		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	
Multinomial	*K*-vector of integer: $[0, N]$	count of occurrences of different types (1, ..., *K*) out of *N* total *K*-way occurrences			

statsmodels.discrete.count_model.ZeroInflatedPoiss

```
class statsmodels.discrete.count_model.ZeroInflatedPoisson(
    endog,
    exog,
    exog_infl=None,
    offset=None,
    exposure=None,
    inflation='logit',
    missing='none',
    **kwargs
)                                                          [source]
```

Poisson Zero Inflated Model

Parameters

endog : array_like

A 1-d endogenous response variable. The dependent variable.

exog : array_like

Regression Analysis :: GLM

```
# GLM: original data
model = sm.GLM(dfy, x_scaled, family =
        sm.families.Gaussian())
result = model.fit()
result.summary(xname=['const']+list(dfx.
        columns))
```

Generalized Linear Model Regression Results

Dep. Variable:	target	**No. Observations:**	442
Model:	GLM	**Df Residuals:**	431
Model Family:	Gaussian	**Df Model:**	10
Link Function:	Identity	**Scale:**	2932.7
Method:	IRLS	**Log-Likelihood:**	-2386.0
Date:	Tue, 04 Feb 2025	**Deviance:**	1.2640e+06
Time:	04:03:32	**Pearson chi2:**	1.26e+06
No. Iterations:	3	**Pseudo R-squ. (CS):**	0.6491
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	152.1335	2.576	59.061	0.000	147.085	157.182
age	-0.4767	2.845	-0.168	0.867	-6.053	5.100
sex	-11.4198	2.915	-3.917	0.000	-17.134	-5.706
bmi	24.7546	3.168	7.813	0.000	18.545	30.964
bp	15.4469	3.115	4.958	0.000	9.341	21.553
s1	-37.7226	19.842	-1.901	0.057	-76.612	1.167
s2	22.7019	16.144	1.406	0.160	-8.940	54.344
s3	4.8116	10.121	0.475	0.634	-15.024	24.647
s4	8.4316	7.689	1.097	0.273	-6.639	23.502
s5	35.7749	8.186	4.370	0.000	19.731	51.819
s6	3.2203	3.142	1.025	0.305	-2.938	9.379

Evaluation Metrics for Regression

- Evaluation metrics for regression are essential tools that help us understand and quantify how well a regression model performs. Here are several reasons why these metrics are necessary:
  - Measure prediction errors,
  - Assess overall model fit,
  - Allow for model comparisons,
  - Guide improvements and tuning,
  - Ensure generalization to new data, and
  - Facilitate communication of model performance.

Evaluation Metrics for Regression

(1) Residual Error Metrics

These metrics are designed to evaluate the discrepancies between predicted values and the ground truth.

Metrics	Mathematical Formula	Definition
Mean Absolute Error (MAE)	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i \rvert$	The average of the absolute differences between predicted and actual values.
Mean Squared Error (MSE)	$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	The average of the squared differences between predicted and actual values.
Mean Absolute Percentage Error (MAPE)	$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left\lvert \frac{y_i - \hat{y}_i}{y_i} \right\rvert$	The average absolute percent error between predicted and actual values. Useful when you need a relative measure of error. MAPE can be problematic if any actual values y_i are zero.
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	The square root of the MSE, providing error in the same units as the target variable.

Evaluation Metrics for Regression

(2) The Proportion of Variance Predicted

These two metrics are used to measure the proportion of variance explained or captured.

Metrics	Mathematical Formula	Definition
R-squared (Coefficient of Determination)	$R^2 = 1 - \frac{SSE}{SST}$ $= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	Measures the proportion of variance in the dependent variable that is predictable from the independent variables.
Log-likelihood	$logL(\beta, \sigma^2)$ $= -\frac{n}{2}\log(2\pi\sigma^2)$ $-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Measures how probable the observed data are under a given model. For regression models assuming normally distributed errors, the log-likelihood is derived from the probability density of the normal distribution. A higher (less negative) log-likelihood indicates a model that better explains the observed data.

Evaluation Metrics for Regression

(3) The Suitability of Predictors

These metrics are developed to quantify the suitability of the predictor set in a model.

Metrics	Mathematical Formula	Definition
Adjusted R-squared	$adj.R^2 = 1 - \left(\frac{1-R^2}{n-p-1}\right)(n-1)$	A modified version of R-squared that adjusts for the number of predictors in the model, providing a more accurate measure when comparing models with a different number of predictors.
Akaike information criterion (AIC)	$AIC = -2logL + 2k$	AIC provides a measure for model selection that balances model fit and complexity. It penalizes models with more parameters (k, inc. constant) to avoid overfitting. Lower AIC values indicate a better trade-off between goodness of fit and model complexity.

Evaluation Metrics for Regression

(3) The Suitability of Predictors

These metrics are developed to quantify the suitability of the predictor set in a model.

Metrics	Mathematical Formula	Definition
AICc	$AICc = AIC + \frac{2k(k+1)}{n-k-1}$	AICc is a small-sample correction to AIC, which is particularly useful when the sample size n is not large relative to the number of parameters p. When n is small, AICc provides a more accurate estimate of the information loss by adjusting AIC's bias toward more complex models.
Bayesian information criterion (BIC)	$BIC = -2logL + klog(n)$	BIC (Schwarz Information Criterion) is similar to AIC but imposes a heavier penalty for models with many parameters, especially as the sample size increases. Lower BIC values suggest a better model. Due to the $k\log(n)$ penalty term, BIC tends to favor simpler models more strongly than AIC, especially with larger datasets.

Evaluation Metrics for Regression

(4) Overall Model Test

These statistical tests are used to determine whether the model meets the underlying assumptions of regression analysis, such as F-test, Durbin-Watson, Omnibus, Jarque-Bera, Degrees of Freedom, Number of Observations, skewness, and kurtosis.

The first four hypothesis tests evaluate the fitted model's goodness of fit, and the other two indicators are the descriptive statistics of the fitted model.

Evaluation Metrics for Regression

(4) Overall Model Test

(a) F-test

The F-test in regression analysis is used to assess the overall significance of the regression model. In simpler terms, it tests whether the model — with all its predictors — provides a significantly better fit to the data than a model with no predictors (i.e., a model that includes only an intercept).

Evaluation Metrics for Regression

(4) Overall Model Test

(a) F-test

Null Hypothesis (H_0): All the slope coefficients (excluding the intercept) are equal to zero. This implies that none of the predictors have any linear relationship with the dependent variable.

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_k = 0$$

Alternative Hypothesis (H_1): At least one of the slope coefficients is not zero. This suggests that at least one predictor contributes to explaining the variability in the dependent variable.

$$H_1: At\ least\ one\ \beta_j \neq 0$$

Evaluation Metrics for Regression

(4) Overall Model Test

(a) F-test

F-statistic is defined as

$$F = \frac{Mean\ Square\ Regression\ (MSR)}{Mean\ Square\ Error\ (MSE)} = \frac{\left(\frac{SSR}{k}\right)}{\left(\frac{SSE}{n-k-1}\right)}$$

$$= \frac{\frac{1}{k}\sum_{i=1}^{n}(\hat{y}_i - \bar{y}_i)^2}{\frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, where\ SST = SSR + SSE$$

SSR (Sum of Squares Regression): Measures the variation explained by the model.
SSE (Sum of Squared Errors): Measures the variation not explained by the model.
k: Number of predictors.
n: Total number of observations.
MSR: SSR divided by k.
MSE: SSE divided by the residual degrees of freedom $(n-k-1)$.

Evaluation Metrics for Regression

(4) Overall Model Test

(b) Durbin-Watson (DW) Test

It is used in regression analysis to detect the presence of autocorrelation (serial correlation) in the residuals (errors) from a regression model. Autocorrelation means that the residuals are correlated across observations, which is a common issue, especially in time series data.

Detecting Autocorrelation:

The DW test examines whether the residuals from your regression are independent of each other. If residuals are autocorrelated, it violates one of the key assumptions of ordinary least squares (OLS) regression (i.e., that the errors are uncorrelated). This can affect the reliability of statistical tests and confidence intervals.

Evaluation Metrics for Regression

(4) Overall Model Test

(b) Durbin-Watson (DW) Test

DW statistics (D):

$$D = \frac{\sum_{t=1}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2}, D \in [0,4], \begin{cases} \approx 2, & no\ autocorrelation \\ < 2, & positive\ autocorrelation \\ > 2, & negative\ autocorrelation \end{cases}$$

where e_i is the residual at time t; n is the numer of observations.

Evaluation Metrics for Regression

(4) Overall Model Test

(c) Omnibus Test

The Omnibus test examines whether the residuals (errors) of the regression model depart from a normal distribution. It does this by jointly testing for deviations in both **skewness** (asymmetry) and **kurtosis** (tailedness).

Instead of testing skewness and kurtosis separately, the Omnibus test combines these two measures into a single test statistic. Under the **null hypothesis** that the residuals are normally distributed, the test statistic follows (approximately) a χ^2 distribution with 2 degrees of freedom.

Evaluation Metrics for Regression

(4) Overall Model Test

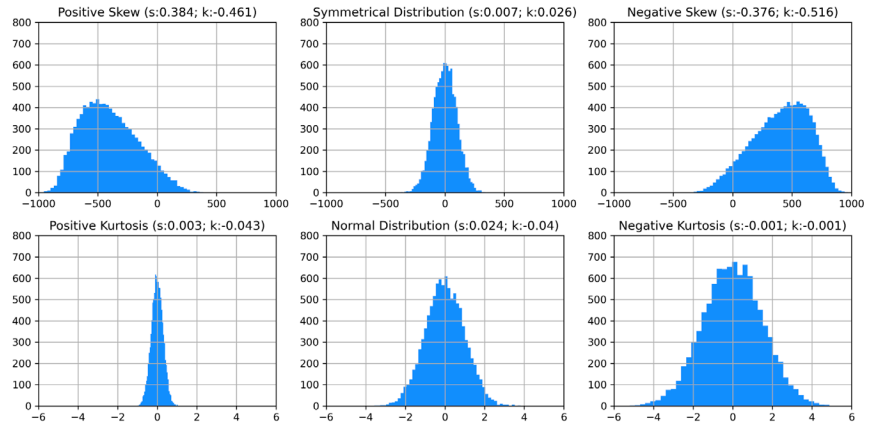
(c) Omnibus Test

The Omnibus test for normality (often implemented in packages like statsmodels as part of the regression diagnostics) is based on combining measures of skewness and kurtosis into a single test statistic. A common version is the D'Agostino-Pearson omnibus test. Its test statistic is defined as:

$$K = Z_1^2 + Z_2^2$$

Z_1 is the *z*-score of skewness

Z_2 is the *z*-score of kurtosis



Evaluation Metrics for Regression

(4) Overall Model Test

(c) Omnibus Test

Diagnostic Tool: The Omnibus test is one of several diagnostics (along with the Jarque-Bera test, Q-Q plots, etc.) that help you understand whether the residuals behave as expected under the model assumptions.

Model Validity: Non-normal residuals, as indicated by a significant Omnibus test, might prompt further investigation into model specification, transformation of variables, or the use of robust standard errors.

Central Moment

The moments of a function are certain quantitative measures related to the shape of the function's graph.

$$\mu_n = \int_{-\infty}^{\infty} (x - c)^n f(x) dx$$

- **1st central moment:**

$$\mu \equiv \mathbb{E}[X] = \int_{-\infty}^{\infty} x f(x) dx$$

- **2nd central moment:**

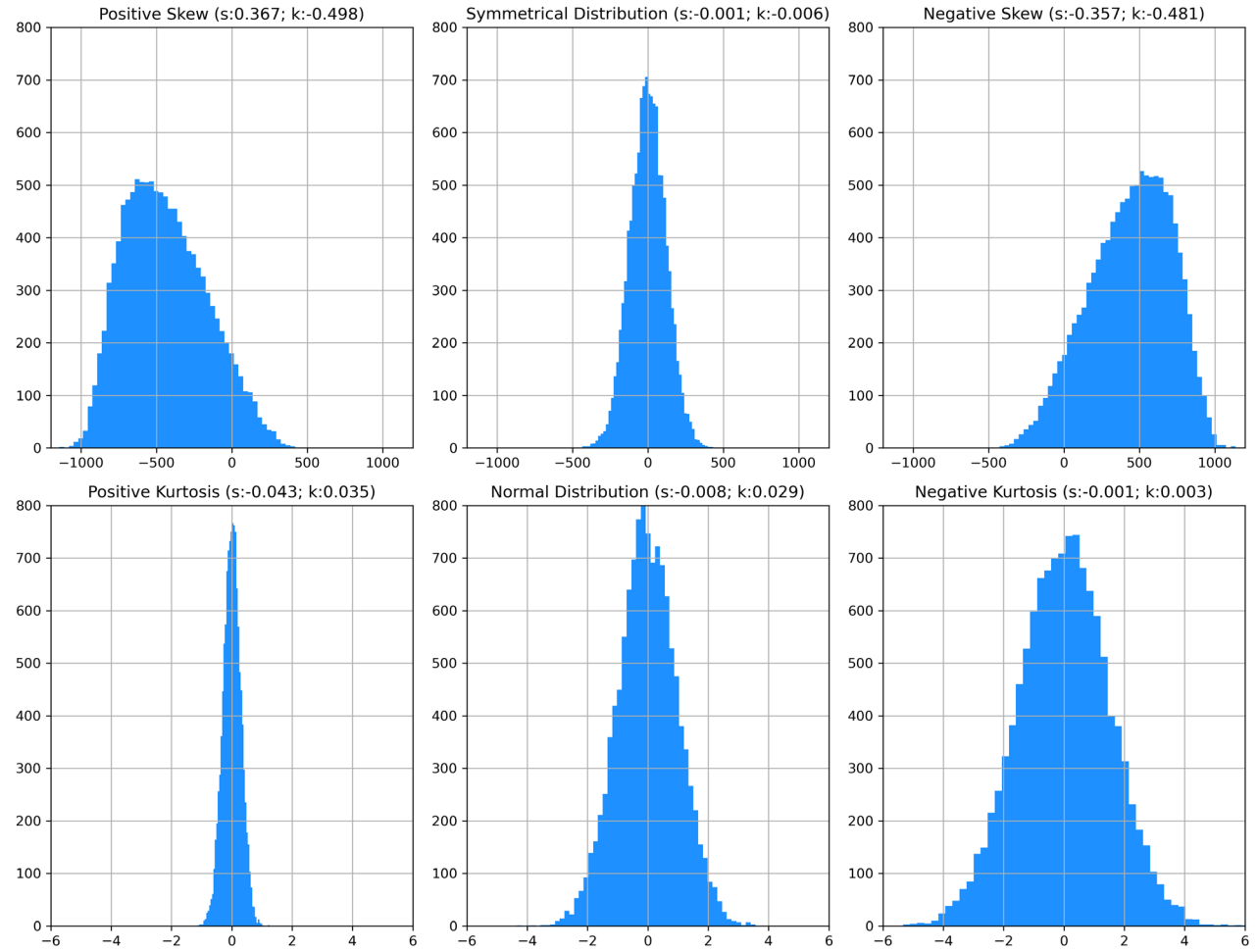
$$Var(x) = \sigma^2 \equiv E(x - \mu)^2 = \int_{-\infty}^{\infty} [x - E(x)]^2 f(x) dx$$

- **3rd central moment:**

$$\int_{-\infty}^{\infty} [x - E(x)]^3 f(x) dx$$

- **4th central moment:**

$$\int_{-\infty}^{\infty} [x - E(x)]^4 f(x) dx$$



Evaluation Metrics for Regression

(4) Overall Model Test

- **Deviance** is a measure of goodness-of-fit for a regression model, especially in the context of generalized linear models (GLMs). It quantifies how well a model explains the observed data by comparing it to a "saturated" model — a model that fits the data perfectly.

$$\begin{cases} linear\ regression & D = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ logistic\ regression & D = 2\sum_{i=1}^{n} \left[y_i \log\left(\frac{y_i}{p_i}\right) + (1 - y_i) \log\left(\frac{1 - y_i}{1 - p_i}\right) \right] \end{cases}$$

Evaluation Metrics for Regression

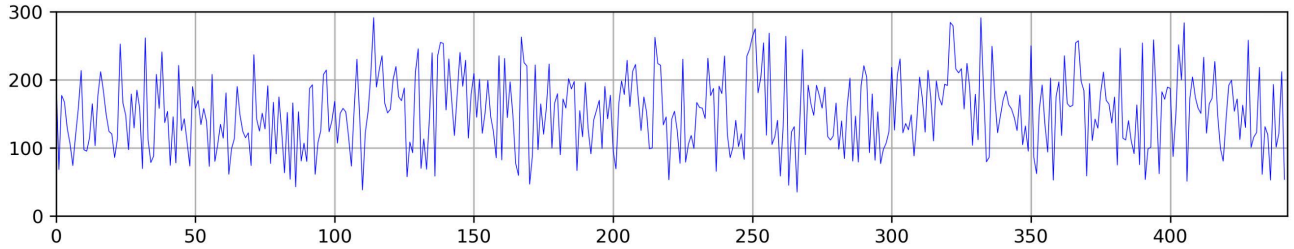
(4) Overall Model Test

- In regression analysis — especially within the framework of generalized linear models (GLMs) — the **Pearson chi-square statistic** is used as a measure of goodness-of-fit. It essentially quantifies how well the model's predictions match the observed data by comparing the observed values with the expected values under the model.
- Large χ^2: May indicate lack of fit if χ^2 significantly exceeds the expected range for that DF.
- Small χ^2: Suggests the model fits well (though always check other diagnostics too).

Assignment #01

```
# Format
def myOLS(xData, yData):
    # annotation
    ...
    ...
    ...
    return y_hat, dfs
```

- You may use the diabetes dataset to test your **myOLS** function.
- Here is the distribution of **y_hat**.



- The result of **dfs** is as follows.

	variables	beta_hat	std_err	t_stats	p_val	ci_lower	ci_upper
0	const	152.1334841628959	2.5728714447608194	59.12984283481704	0.0	147.076581215127	157.19038711066483
1	age	-0.4766602999910251	2.8419072492758244	-0.16772549495149353	0.866877726904435	-6.062345187361987	5.109024587379937
2	sex	-11.419792555829664	2.9119735092894286	-3.921667734750887	0.00010225444957123742	-17.143190610923575	-5.696394500735751
3	bmi	24.75456762164087	3.1645901819659445	7.822361253191572	4.0190073491430667e-14	18.53465894635458	30.97447629692716
4	bp	15.44688788106301	3.11172515650134	4.964091333320285	9.951839481381342e-07	9.330883854277605	21.562891907848414
5	s1	-37.722649454866975	19.81892012181858	-1.9033655326829964	0.05765723265448974	-76.67615247178051	1.230853562046569
6	s2	22.70185814310681	16.125612908158644	1.4078136609381813	0.1599054179825803	-8.992558465193184	54.39627475140681
7	s3	4.811584187525156	10.10882517725403	0.475978573489603	0.6343302576716332	-15.05701357760292	24.680181952653232
8	s4	8.431582746254701	7.680418744644037	1.0978024801231692	0.27290225083471475	-6.664053792049193	23.527219284558598
9	s5	35.77493807414732	8.17623373044388	4.37547888839586	1.5209147055550432e-05	19.70479180340232	51.84508434489233
10	s6	3.22031867541451	3.138469855409411	1.0260792117738604	0.3054284378040708	-2.9482512683430855	9.388888619172105

Assignment #01

- Here are some tips for matrix calculation.

```
# β̂ = (X^T X)^{-1} X^T y
XTX = x_scaled.T @ x_scaled
XTy = x_scaled.T @ y
# compute β̂
beta_hat = np.linalg.inv(XTX) @ XTy
```

- The t-distribution could be obtained from package.

```
from scipy.stats import t
t_crit = t.ppf(1 - alpha/2, df=df_error)  # critical t value
```

Assignment #02

```
# Format
def myEva(xData, yData, yEst.):
    # annotation
    ...
    ...
    ...
    return RMSE, MSE, MAE
```

- You may use the diabetes dataset to test your **myEva** function.

The End

Thank you for your attention!

Email: chchan@ntnu.edu.tw

Website: https://toodou.github.io/